Bridging the Gap between Privacy and Security in Multi-Domain Federations with Identity Tokens

David J. Lutz
Computing Center
Universitaet Stuttgart
Stuttgart, Germany
Email: David.Lutz@rus.uni-stuttgart.de

Ruth del Campo
Computing Center
Universitaet Stuttgart
Stuttgart, Germany
Email: Ruth.delCampo@rus.uni-stuttgart.de

***Abstract*— In this paper we introduce the concept of Identity Tokens in multi-domain service environments and show how it is used to bridge between authentication/authorization and users' privacy. This token provides, together with further authentication and authorization techniques, a high level of privacy without anonymity.**

I. Introduction

It's a challenge to bring privacy and security[1] together. Privacy requires as little information as possible about a user, but, for for security, it is important to receive all relevant information.
We see a Single-Sign-On (SSO) future where a user wanting to access resources on a network has to login only once regardless of which domain he uses for access. But with SSO, an authentication/authorization problem occurs due to the non-centralized structure of large networks, when the user wants to access services in another domain other than his Home Domain. From the user's point of view, it is desirable not to provide any private information concerning his real identity or contractual information to a domain he does not trust. However, the domain owner needs access to this information for charging him when he uses a service. We assume that all important private information is stored at the Home Domain of the user. He has a contract with its owner, which he trusts, and sees no risk in providing this information, but he is doubtful in doing so to Foreign Domains. So we need an approach that guarantees privacy but provides non-repudable security information to authenticate and authorize the user.

A. Domain

In this paper a domain is seen as a network of different components that are owned and administrated by one person or organization which wants to earn money or other benefits from providing users access to the domain. We assume that in each domain there is at least

- one access point through which users can access the domain
- one A4C[2] System (see below) that handles authentication, authorization, accounting, charging and auditing and
- one service (maybe requiring payment)

Underlying network technologies, domain owner and administration tools, etc. are also part of a domain, but the above mentioned parts are the essentials for our scenarios.
With the term Home Domain (HD), we identify the domain a user has a contract with; the A4C system of the HD stores private information about its users e.g. billing details, address, etc. and the domain owner charges the user (in this context, the customer) for the services he has consumed in his Home Domain and any collaborative Foreign Domains.
The term Foreign Domain (FD) means any domain that is not the user's Home Domain. In that case, we assume that the user does not want to share his private information with the owner of that (from his point of view untrusted) domain.

B. Multi-domain-Federations

Multi-domain-Federations (MDFs) occur in large networks, where services and access points are distributed over more than one domain. We propose that the owners of different domains in an MDF have contracts and trust each other, but that they have their own systems for A4C. Services provided in a domain should also be offered to outside users to increase earnings.
Using such an MDF with SSO means dealing with problems in the authenticate/authorize phase. When a user from another domain wants to access a service in an FD, he has to authenticate himself for authorization purposes. However, it is obvious that he should not have to provide all his accounting-data to each domain. So the problem occurs in how a user can be authenticated/authorized himself in such an FD without giving away private information. Often, it is intended that the user uses a specific profile in a domain that he can configure, so there is another issue to consider. Although the user does not want to provide any private information, he wants to use the specific profile he has developed for that domain and that is explicitly associated with him.

C. Protocols

We use the Security Assertion Markup Language (SAML) [1] for exchanging security and privacy information in the form of assertions. Three kinds of assertions, provided by SAML are used: Authentication Assertions, Authorization

[1]We focus here only on that aspect of security that the identity of the user can be detected definitely, so that it is certain which user consumes a service.
[2]authentication, authorization, accounting, charging, auditing

0-7803-9792-4/06/$20.00 ©2006 IEEE

Assertions and Attribute Assertions.
These assertions are responses to specific requests and provide the available information about authentication, authorization and associated attributes. Additionally, SAML is used to generate the IDToken during the login phase.

Diameter [2] is an AAA (authentication, authorization and accounting) protocol for applications such as network access or IP mobility and it is used by us to exchange messages between most components and the A4C System with Attribute-Value Pairs (AVP's). The Diameter protocol is not bound to a specific application running on top of it (e.g. Mobile IP, SIP or NAS applications), but focuses on general message exchange features. Because authentication and authorization mechanisms vary among applications, the Diameter Base Protocol does not define command codes or AVPs specific to authentication and authorization.

II. A4C System

In our approach all tasks regarding authentication, authorization, accounting, charging and auditing in one domain are handled by an A4C System consisting of at least one A4C Server along with A4C clients and one SAML Authority [3]. The A4C Server stores all relevant information about the user. In the HD it is the only component that has access to knowledge about the user's real identity, the charging information and other contractual data while in an FD even its A4C System is not aware of this data. Furthermore, the Digital Identity (profile, attribute list) of the user is also stored on the A4C System. We focuses here only to that part of the A4C System that is responsible for the IDToken and the tasks associated with it, i.e. the SAML Authority.

III. SAML Authority

The SAML Authority is part of the A4C System, but does not have to be part of an A4C Server. In that case it communicates with the A4C Server using a SAML-SOAP bindig [1], [4].
In our approach, the SAML Authority is responsible for generating the SAML Assertions and IDTokens. The IDTokens are stored together with authentication information such as username, kind of authentication, etc. in a database for verifying the user and creating SAML Authentication Assertions when requested. There are three main cases whith which the SAML Authority is involved:

- Login
  When the user logs-in, the SAML Authority of that domain is requested by the A4C System to generate an IDToken for him. If required, the user can develop a profile, thus providing some attributes that are visible to this domain or he can choose a previously stored profile.
- Authentication
  When a new authentication of the user is required, the SAML Authority receives his IDToken in the request via the A4C System and builds a SAML Authentication Assertion providing the authentication information that is assigned to the user if the token is valid. Due to the static organisation of the A4C System (there are no sudden changes to the components or their addresses) all parties in that domain can trust the information in the Assertion.
- Authorisation
  If the user wants to access a resource, the decision to grant access is made at the Policy-Management-System (PMS) of that ressource by authorization information. The PMS requests the user's attributes from the SAML Authority (using the A4C System) by sending the user's IDToken to the Authority. If the token is valid it responds with an Attribute Assertion.

Thus, the SAML Authority is the main component for handling the IDToken and for providing the user's digital identity (i.e., his chosen profile) to a domain.
The concept of a centralized A4C system in each domain with assigned authorities has also the advantage that due to the trust within a domain and the contracts between domain owners, a Certificate Authority can be attached to the A4C. In that case, it is easy to build a PKI system within the whole federation so that all important messages can be encrypted or digitally signed.

IV. Identity Tokens

When we think of an MDF, we and assume that the user wants to be authenticated and authorizated in an FD using some kind of mobile terminal. We solved the problems of security and privacy by providing a small Identity Token (IDToken) that will be stored on the user's mobile terminal. After the user logs-in successfully, he receives the IDToken from the SAML Authority of that domain via the A4C System. Each time the user has to seek authentication (assuming he has already logged-in) or to seek authorization, he sends his token to the requestor from where it will be redirected to the A4C Server, in the current domain, using the Diameter protocol. The A4C Server forwards the token to the SAML Authority to have it validated and the requestor receives the Authority's response via the A4C System.
The IDToken is a string built from the following components:

- SAML Artifact
  This item is a pointer to the SAML Authentication Assertion in the SAML Authority.
- Serial Number
  The Serial Number is used as a counter and is increased each time the ID Token is used.
- Random Number
  The random number changes to avoid replay attacks each time the IDToken is used.
- Signature
  The signature of the issuer of the token, i.e., the A4C System when first issued and the mobile terminal of the user following the login-phase.

Each time the user uses his IDToken, it must first be updated. That means: increment the serial number and generate a new random number. These changes made for every request make

this approach more secure than a static token.
There are three main cases that involve handling the IDToken: Login, Authentication and Authorization.

Login to a Foreign Domain
Firstly, the user seeks authentication himself using the foreign A4C for SSO by sending his username and encrypted password. The foreign A4C recognizes where the authentication information is stored (e.g., via a certificate or a simple WAYF) and requests authentication information from the home A4C. After validating the user via his home A4C, the foreign A4C requests the SAML Authority (in the FD) to generate an IDToken for him, that is then sent by the foreign A4C to the (mobile) terminal of the user where it is stored. This IDToken is combined with the profile of that user in the FD. The SAML Authority uses the IDToken as a pointer to the user's attribute list. In that way, the home A4C is able to match his profile in the FD with his real identity and his charging information (no anonymity), but the user in the FD is only visible with the profile that he created for that domain (pseudonymity). Thus, it is guaranteed that no private information is stored on the foreign A4C. This case is shown schematicly[3] as a sequence diagram in Fig. 1.
If the user logs-in to his home domain, the steps for getting the IDToken are the same but without the connection between the foreign A4C and home A4C.

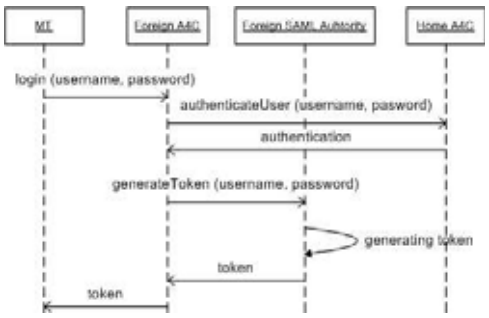

Fig. 1. Login in a MDF

Authentication within a domain
Each time the user needs to seek authentication after the login-phase, he updates his Identity Token and sends it together with his username to the requestor. The requestor contacts his A4C asking if the IDToken matches the username. If so, the SAML Authority sends back an authentication assertion.
If the service does not belong to the same domain as the A4C that issues the IDToken, the A4C of the service domain must initially contact the A4C of the login domain.

[3]Not all steps are identified in the sequence diagrams. Intermediary components that are not important in understanding Identity Tokens are not shown. The commands have also been simplified.

Authorization in a Foreign Domain
When the user wants to access a special service provided by the FD, he first tries to seek authorization by sending his IDToken to the Policy-Management-System (PMS), which then tries to obtain authorization information by forwarding the token to the A4C of that domain. The SAML Authority, after validating the token, generates an Attribute Assertion and sends it back to the PMS where the user's request is evaluated (see Fig. 2).
In this case, the privacy of the user is secured twice, since the private credentials of the user are not visible to the FD and the service is not even aware of the user's profile, but only of the decision of the PMS. Due to the Identity Token-Concept the authorization is guaranteed and the user can be charged for consuming the services according to the contract he has with his HD and the contract between his HD and the FD.

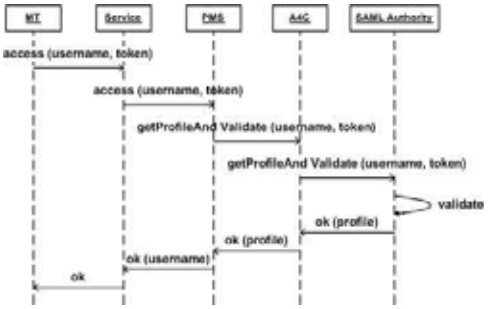

Fig. 2. Authorization in Multi-domain Networks

V. Conclusion

The Identity Token is one approach to providing privacy in FDs without losing the ability for authentication, authorization, accounting, charging and auditing. It is a useful tool in multi-domain federations that helps to bridge the gap between privacy and security/authentication and so offers the chance to grant services requiring payment to a user without forcing him to give his private credentials to an unknown domain.
This approach fulfills the requirements for operating in mobile networks and has been successfully tested in AKOGRIMO [3].

Acknowledgment

These results published in this paper were developed in the project AKOGRIMO [3] funded by the EC under the FP6-IST programme. The Authors would like to thank all the partners involved in that project.

References

[1] Security Assertion Markup Language (SAML) - OASIS Standards
http://www.oasis-open.org/specs/index.php
[2] Diameter Protocol
http://www.diameter.org/
[3] "Access to knowledge through the Grid in a Mobile World" (AKOGRIMO) funded by the EC under the FP6-IST programme
http://www.mobilegrids.org/
[4] Simple Object Access Protocol (SOAP)
http://www.w3.org/TR/soap/